Hindawi
Journal of Sensors
Volume 2021, Article ID 6598782, 13 pages
https://doi.org/10.1155/2021/6598782

Research Article

Analysis of the Distributed Immune Inspection Equipment Sensor Scheduling Model Based on Adaptive Dynamic Probabilistic Particle Swarm Optimization

Yi Zhou,[1] **Weili Xia**,[1] **and Shengping Peng**[2]

[1]*School of Management, Northwestern Polytechnical University, Xi'an 710072, China*
[2]*School of Economics and Management, Three Gorges University, Yichang 443002, China*

Correspondence should be addressed to Shengping Peng; pengshengping@ctgu.edu.cn

Received 14 October 2021; Accepted 29 November 2021; Published 16 December 2021

Academic Editor: Guolong Shi

Based on the analysis of bacterial parasitic behavior and biological immune mechanism, this paper puts forward the basic idea and implementation method of an embedding adaptive dynamic probabilistic parasitic immune mechanism into a particle swarm optimization algorithm and constructs particle swarm optimization based on an adaptive dynamic probabilistic parasitic immune mechanism algorithm. The specific idea is to use the elite learning mechanism for the parasitic group with a strong parasitic ability to improve the ability of the algorithm to jump out of the local extreme value, and the host will generate acquired immunity against the parasitic behavior of the parasitic group to enhance the diversity of the host population's particles. Parasitic behavior occurs when the number of times reaches a predetermined algebra. In this paper, an example simulation is carried out for the prescheduling and dynamic scheduling of immune inspection. The effectiveness of prescheduling for immune inspection is verified, and the rules constructed by the adaptive dynamic probability particle swarm algorithm and seven commonly used scheduling rules are tested on two common dynamic events of emergency task insertion and subdistributed immune inspection equipment failure. In contrast, the experimental data was analyzed. From the analysis of experimental results, under the indicator of minimum completion time, the overall performance of the adaptive dynamic probability particle swarm optimization algorithm in 20 emergency task insertion instances and 20 subdistributed immune inspection equipment failure instances is better than that of seven scheduling rules. Therefore, in the two dynamic events of emergency task insertion and subdistributed immune inspection equipment failure, the adaptive dynamic probabilistic particle swarm algorithm proposed in this paper can construct effective scheduling rules for the rescheduling of the system when dynamic events occur and the constructed scheduling. The performance of the rules is better than that of the commonly used scheduling rules. Among the commonly used scheduling rules, the performance of the FIFO scheduling rules is also better. In general, the immune inspection scheduling multiagent system in this paper can complete the prescheduling of immune inspection and process dynamic events of the inspection process and realize the prereactive scheduling of the immune inspection process.

1. Introduction

The development of smart sensors in recent years has laid the foundation for the advancement of wireless sensor networks (WSN) [1]. Wireless sensor networks are composed of miniature sensors that can monitor physical and environmental factors (such as temperature, humidity, vibration, motion, and seismic events). The sensor node is small in size, low in price, and intelligent. The emergence of Internet of things (IoT) has increased the range of requirements for wireless sensor networks and further expanded the ongoing research in this field. IoT is a network that interconnects smart objects through communication media. Sensor nodes are the main components of IoT. Therefore, WSN is expected to play an important role in IoT. By 2022, it is estimated that 50 billion distributed immune inspection equipment will be connected to the network, most of which will be equipped with sensors and actuators. Therefore, wireless sensor networks will be critical to the effectiveness of the Internet of things [2]. WSN consists of sensor nodes deployed in a structured or unstructured manner in selected areas of interest. A typical sensor node consists of a power unit, a radio unit, a monitoring unit, and a processing unit. Due to the limitations of distributed immune inspection equipment, these nodes in the network present several challenges, namely, limited computing power, energy, data storage, and communication bandwidth [3, 4].

The network management in wireless sensor networks is very complicated. The challenge mainly comes from the solidification of the traditional infrastructure network structure, which makes configuration and maintenance very difficult [5]. The simplicity and scalability of software-defined networking (SDN) can simplify its network management. In WSN, if management is not flexible, the reconfiguration and maintenance of sensor nodes are often a complicated and lengthy process. The harsh environment of deploying wireless sensor networks will further aggravate this situation [6]. By introducing SDN, the control logic can be removed from the sensor nodes, making them only serve as forwarding units. These forwarding units will be controlled and operated by a centralized controller, thereby realizing the programmability of physical infrastructure nodes. SDN can also dynamically map configuration between sensor nodes and controllers [7]. Mobility and localization are essential to achieve better deployment of wireless sensor networks. Depending on the nature of the deployment, there may be distributed immunity testing equipment mobility and network mobility. Generally, traditional routing protocols will periodically update the routing table (proactively) or request routing in other ways when the network changes. This process is energy-intensive and not suitable for WSN networks. SDN simplifies this process through mobility management from a central controller; that is, the controller manages routing decisions and policies. Localization algorithms can also be implemented at the controller or application level, instead of sensor nodes with resource constraints [8]. Traditional wireless sensor networks deployed for specific tasks have the problems of insufficient utilization of network resources and imbalanced energy consumption of nodes. The main reason is that multiple sensor networks of different suppliers are independently deployed in the same monitoring area and have mutual resources [9]. In addition, traditional wireless sensor networks rely too much on proprietary services, lack the flexibility to implement instant changes, and cannot respond in time and take effective measures in the face of dynamic changes in the network topology [10]. Therefore, how to optimize network performance through flexible and efficient node and network resource scheduling strategies when node resources are limited and the network changes dynamically has important research significance.

In order to improve the energy efficiency of the network, this paper establishes an energy model based on node hardware and proposes an energy-optimized sensor scheduling strategy, which can balance the energy consumption of nodes while ensuring tracking accuracy. Specifically, the technical contributions of this article can be summarized as follows.

First, this article analyzes the bacterial parasitic behavior and biological immune mechanism in the development of biological parasitic mechanisms and then embeds the biological adaptive dynamic probabilistic parasitic immune behavior mechanism into the particle swarm optimization algorithm and constructs a particle swarm based on an adaptive dynamic probabilistic parasitic immunity algorithm model.

Second, we test the effect of the adaptive dynamic probabilistic parasitic immunity mechanism on the particle swarm optimization algorithm through simulation experiments. This paper verifies the effectiveness of ABC-TLBO for immune inspection prescheduling and simulates two dynamic events of emergency task insertion and subdistributed immune inspection equipment failure during the prescheduling operation of the system.

Third, this article compares the rules constructed by the adaptive dynamic probability particle swarm algorithm with seven commonly used rules. The experimental results show that the performance of the adaptive dynamic probability particle swarm algorithm is better than that of the other seven rules, which proves the effectiveness and practicability of the scheduling rules constructed by the adaptive dynamic probability particle swarm algorithm proposed in this paper.

2. Related Work

Scientific researchers have designed a heuristic intelligent optimization algorithm based on bionics by simulating the behavior and activities of various biological phenomena by referring to various biological phenomena in nature. Different from the traditional optimization method, which is step by step according to the fixed procedure, the heuristic intelligent algorithm based on bionics selects a better optimization method according to the set rules. Heuristic algorithms have few requirements on the mathematical nature of the problem itself, and some do not even require it. This improves the applicability of the algorithm on the one hand and on the other hand greatly reduces the number of searches for large-scale problems and saves search time. For complex optimization problems, a heuristic algorithm is an effective solution tool. The swarm intelligence algorithm constructs a class of self-organizing and self-adaptive stochastic optimization algorithms by simulating group behaviors in nature. There are some special phenomena in the behavior of biological groups in nature. For example, the places where the food concentration is highest are often the places where the fish 5gather; the birds keep a reasonable distance between the individuals during the flight to realize the rapid approach and pursuit of the target; the colony can always find the shortest path between the ant colony and the food. By imitating the behavior mechanism between individuals in a biological group and applying it to the design of an evolutionary computing algorithm, a swarm intelligence optimization algorithm based on biomimetic is formed.

Research on the fusion of particle swarm optimization algorithms with other intelligent algorithms has been gradually launched [11]. Early researchers combined particle swarm optimization algorithms with genetic algorithms and later considered combining them with simulated annealing and chaos theory [12]. The effect of various technologies in the improved algorithm needs further research. The fusion of the particle swarm optimization algorithm with new algorithms such as the ant colony algorithm, fish school algorithm, and frog leaping algorithm is gradually emerging [13]. However, the attempts and research of embedding the biological behavior mechanism into the algorithm to improve the algorithm need to be in-depth.

By introducing SDN into the wireless sensor network, efficient network resource scheduling and flexible network management can be realized, which have become two of the effective solutions to the above-mentioned problems, and the software-defined wireless sensor networks (SDWSN) were born from this [14, 15]. SDWSN transfers the control logic from the sensor node to the logically centralized master node (control server), which realizes the decoupling of the control layer and the data layer and enables ordinary nodes to only have data collection and forwarding functions, improving their work efficiency and reducing energy consumption. The master node has a global network view, which can flexibly schedule network resources according to different task requests, select the lowest activated bottom node to perform multiple tasks at the same time, and solve the problems of unreasonable resource scheduling in traditional wireless sensor networks and low node energy efficiency [16, 17]. In addition, when network dynamic events (nodes join and leave) occur, the master node can learn topology changes in time based on the global network view and then activate or shut down nodes through resource scheduling within the cluster to ensure monitoring quality and reduce energy consumption [18].

Related scholars have proposed a wireless sensor network framework (SDNSense) based on software-defined networks, which separates network control from hardware through software-defined networks to improve network flexibility and dynamically adapt to network changes [19]. Researchers realize the automatic reconfiguration of wireless sensor networks through software-defined networks and propose a green routing algorithm based on the adaptive particle swarm optimization algorithm to maximize network life [20]. Related scholars have proposed a software-defined wireless sensor network architecture (Soft-WSN) for IoT applications, which implements distributed immune inspection equipment management and network management through a software-defined controller to meet the needs of IoT applications [21]. The above-mentioned literature mainly studies the architecture design and implementation of SDWSN, focusing on how to realize the decoupling of wireless sensor networks through software-defined networks. Regarding SDWSN resource scheduling, related scholars have proposed a wireless sensor network sleep scheduling mechanism based on the software-defined network [22]. The controller executes the resource scheduling algorithm to determine the node status, eliminating the two broadcast processes in each cycle to reduce communication energy consumption. However, the flexibility of the scheduling mechanism is insufficient, because the controller only determines whether it is sleeping or not according to the beacon information uploaded by the node at the beginning of each cycle, and cannot modify the node status in real time according to network changes.

Relevant scholars have designed an energy-saving network resource rescheduling strategy in SDWSN [23]. The controller loads different programs into the nodes to dynamically reprogram its functions, so as to efficiently schedule node resources to meet different task requirements. However, in the resource scheduling process, only the energy consumption during rescheduling is considered, and the schedulability of nodes and memory constraints are not considered. Relevant scholars introduced wireless power transmission technology into SDWSN and proposed an optimized energy transmitter placement mechanism and energy-efficient network resource scheduling strategy [24, 25]. However, when the node energy is too low or the network topology changes, the energy transmitter scheduling mechanism needs to be frequently implemented across the entire network, which is not flexible enough, and the rescheduling time is too long.

3. Distributed Collaborative Information Processing Method in the Wireless Sensor Network

3.1. Sensor Scheduling Scheme Based on Least Mean Square Estimated Variance. In the WSN target tracking problem, the state of a moving target can be described by a general state space model:

$$\begin{aligned} X(k+1) &= f[X(k-1), u(k-1)] - w(k-1), \\ Z(k+1) &= h[X(k-1)] - v(k-1). \end{aligned} \tag{1}$$

In the formula, $X(k)$ is the target state, $z(k)$ is the observation vector, $u(k)$ is the input control vector, and $\omega(k)$ and $v(k)$ represent the process noise and observation noise, respectively. The function f connects the state at time $k-1$ in the past with the state at time k, and the nonlinear function h reflects the relationship between the state variable $X(k)$ and the observed variable $z(k)$.

Because the system has no input control quantity, $u(k) = 0$. Since the target is moving on a two-dimensional plane, the state variables of the moving target are

$$X(k) = [x(k-1)\, x_v(k-1)\, y(k-1)\, y_v(k-1)]^T. \tag{2}$$

It represents the state of the target at the k-th sampling period tk, where $x(k)$ and $y(k)$ are the coordinates of the target on the x-axis and y-axis, respectively, and $xv(k)$ and $yv(k)$ are along the x-axis and the y-axis.

We consider the computing power, storage capacity, and energy supply limitations of sensor nodes. In order to achieve the high-precision, high-reliability, and low-latency tracking quality requirements of the tracking system, this paper uses node collaborative information processing to solve this problem. In the implementation process, it is necessary to consider the information perception, data fusion, and scheduling strategy of the node. The collaborative

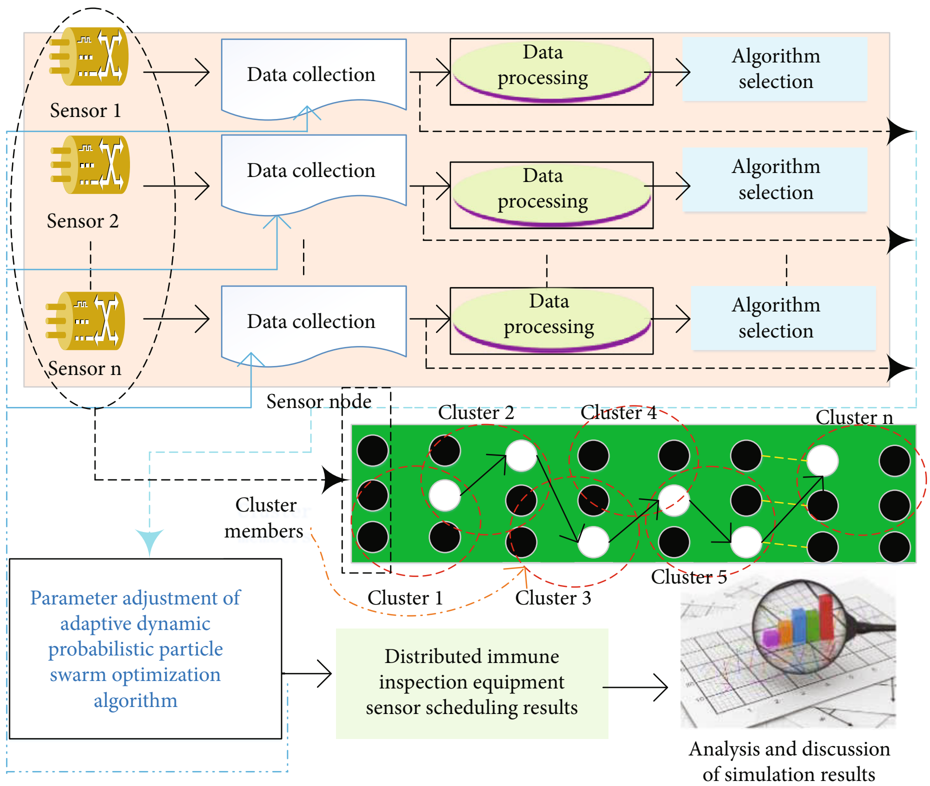

Figure 1: Schematic diagram of clustering of sensor nodes.

computing of sensor nodes can selectively aggregate information from other nodes, improve perception and tracking performance, save communication costs and network resources, and reduce the risk of network node failure.

In order to save energy consumption and communication bandwidth, extend the life cycle of wireless sensor networks, and reduce redundant data of nodes, a dual wake-up/sleep mechanism is also adopted in the actual platform: (1) in the process of node scheduling, only a few are located near the target in the tracking, and the remaining nodes selectively enter the dormant state according to whether to undertake the task; (2) in the target tracking process, the corresponding ultrasonic sensor is controlled by the passive infrared sensor, so that the ultrasonic sensor on the side close to the target is in a usable state. The clustering diagram of sensor nodes is shown in Figure 1.

3.2. Sensor Scheduling Scheme Based on Minimum Mean Square Estimation Variance and Energy Consumption Compromise. From the hardware structure of the tracking platform, energy-consuming components include MICAz nodes, SRF08 ultrasonic sensors, and GH718 passive infrared sensors. Among them, the main energy-consuming components of the MICAz node include a microprocessor, a wireless communication module, a data acquisition board, and a storage unit. Each state of each component is combined to form the state of the entire node. The accuracy of the resulting model depends on the type of components included in the calculation, the workflow, and the execution time. The basic idea of establishing an energy model is as follows:

In the process of node tracking, it is determined which energy-consuming components to operate according to the actual execution of the source code. For the working energy-consuming component, the working time of the component is calculated through the program flow and then multiplied by the power in the corresponding working mode to obtain the energy consumption of the component in the work task process.

Compared with centralized algorithms, distributed collaborative algorithms are more suitable for wireless sensor networks due to the use of modular computing and control, and the data transmission overhead, packet loss, and delay are smaller [26, 27]. The collaborative work between nodes can reduce the risk of single node failure, shorten the tracking sampling period, and have good scalability and reliability.

As the task node, the cluster head node will trigger the ultrasonic ranging. According to the target state at the last moment and the current node's measurement value, the state information of the target is updated through EKF and the state information is broadcasted, and the client computer can receive and display the corresponding trajectory of the movement [28, 29].

Through platform experiment data, it can be obtained that the trace of the state error covariance matrix after the nodes in the cluster run the EKF prediction algorithm is usually between 15 and 25. Through simulation experiments, a rough distribution of the remaining energy of the nodes can be obtained, and the scale coefficients Ke and θ can be set appropriately. We set the waiting return time t_w of the

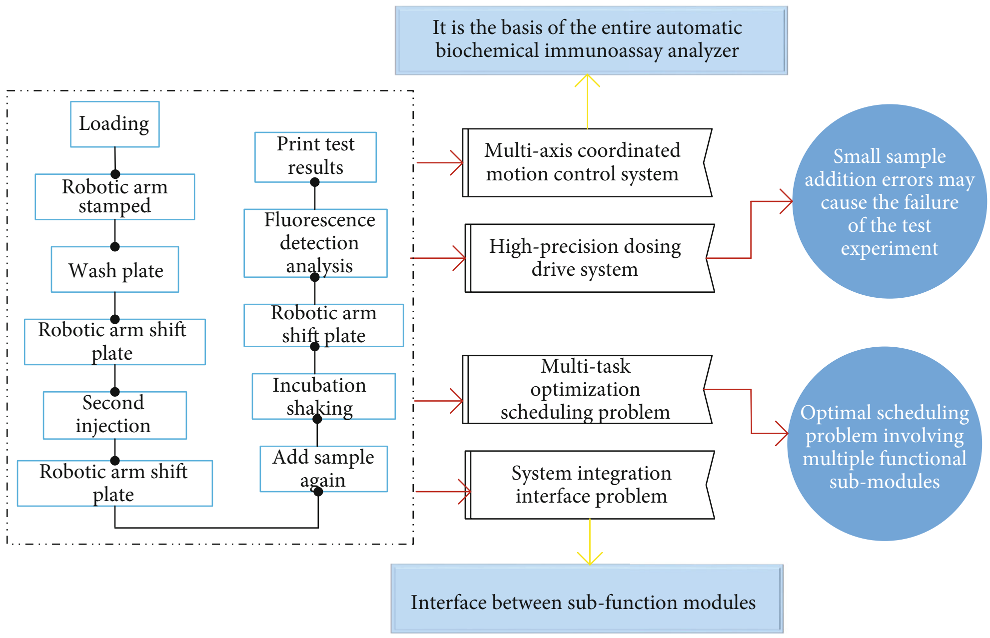

Figure 2: Working process of automatic biochemical immunoassay equipment.

cluster head node to 80 ms according to t_r, and the sampling period T_s of the sensor node is about 130–220 ms. Since this platform is a variable sampling period system, and some random errors will be introduced through hardware characteristics, such as packet loss, time delay, and measurement noise. This platform has certain requirements on the size and form of the moving target speed. The working process of the automatic biochemical immunoassay equipment is shown in Figure 2.

4. Particle Swarm Algorithm Based on the Adaptive Dynamic Probability Parasitic Immune Mechanism

4.1. Analysis of the Adaptive Dynamic Probability Parasitic Immunity Mechanism. The evolution of parasitic relations originates from the spatial and food connections between organisms, which are mainly manifested in three ways: from spatial connection to food connection, transition to parasitism through predation, and organisms accidentally sneaking into the body. The spatial connection between organisms first forms a simple symbiosis in space, gradually transitions to the host body, and finally enters the host body to symbiosis. The formation of the symbiosis relationship provides the basis for the nutrient connection between organisms. A symbiotic relationship of partial benefit is formed that is beneficial to one party and harmless to the other. The host relies on the host to maintain a parasitic relationship, and the host has mutually beneficial symbiotic relationship with each other. The transition to parasitism through predation behavior is manifested as whole ground swallowing, temporary attack and blood sucking, and living on the host body except for the breeding period. The host accidentally sneaked into the host body, the host body is the host's temporary living place, and the host is a facultative parasite. This symbiotic relationship is very beneficial to the survival of the parasite.

The three parasitic relationships produced through symbiosis, predation, and accidental parasitism will evolve in different directions. In the long-term coevolution process, the parasite and the host gradually realize mutual adaptation through mutual struggle, so that the harmful negative effects are weakened. The relationship of mutual benefit and symbiosis has formed two different degrees of immune mechanisms: innate immunity and acquired immunity. Acquired immunity means that the host activates the host's immune system to kill the parasite after the parasite invades and has the ability to resist reinfection with the same parasite. Acquired immunity is widespread in nature. For example, vertebrates in higher animals will produce an immune memory after being infected by parasites. When they are attacked by the same pathogen again, the vertebrates will produce corresponding antibodies to fight the infection. Although animals and plants do not have the complex specificity of vertebrates, they also have a certain degree of acquired immunity. This phenomenon has been found in tobacco plants. When a tobacco leaf is infected by tobacco mosaic virus, the level of chemical substances in the whole plant becomes higher, which improves the ability to resist pathogens.

4.2. Particle Swarm Optimization Algorithm. As a distributed optimization algorithm, the search process of the particle swarm optimization algorithm simulates the bionic movement of the boids group. Individuals move according to the set rules, and the emergence of individual behaviors is expressed as group behaviors. In order to improve the performance of the particle swarm optimization algorithm, a certain information exchange and information sharing mechanism needs to be adopted between individuals in the population. The individual in the particle swarm optimization algorithm needs to be iterated by comparing the individual extremum and the global extremum, so as to keep the population close to the optimal solution as a whole to achieve the goal of optimization.

The particle swarm optimization algorithm also shows obvious swarm intelligence characteristics: first, the calculation of the particle swarm's spatial position in the swarm will change with time; the second is the performance of the particle swarm, which is reflected in every particle swarm optimization algorithm. The third is that the diversity of the population is realized through a specific allocation method, and the specific method is to memorize the optimal particle of the individual and learn the global optimal particle. Fourth, stability and adaptability are reflected in the change of the optimal particle in the population, and the behavior of the particles in the population will change accordingly.

In the particle swarm optimization algorithm, in order to realize the complex behavior of the entire particle swarm, individuals are represented by volumeless and massless particles, and simple behavior rules are set for the particles. The biological principle of the particle swarm optimization algorithm comes from the simulation of bird predation behavior, and the optimal swarm goal depends on the mutual cooperation between birds. In the particle swarm optimization algorithm, each particle is a candidate solution. Many particles are optimized through cooperation. In the process of searching for the optimal solution, individual particles can use their own experience and the best experience in the group to find a better solution in the search space.

The particles in the particle swarm optimization algorithm search for the optimal position by searching for two "extremums" and update themselves according to the obtained information. One is the individual extreme value, that is, the optimal position found during the particle's own optimization process, and the other is the overall extreme value, that is, the optimal position found within the particle domain. According to the different compositions of the particle domain, we divide the particle swarm optimization into a local version and a global version. The former is composed of some particles, and the latter is the domain of the entire population. There are two ways to compose neighborhoods in the local version. One is to compose neighborhoods by particles adjacent to the index number, and the other is to compose neighborhoods by particles that are adjacent in position. The topological structure of the particle swarm refers to the neighborhood definition strategy of the particle swarm optimization algorithm.

4.3. Adaptive Dynamic Probability Parasitic Immune Particle Swarm Optimization Algorithm. The particle swarm optimization algorithm simulates the flight and predation behavior of a flock of birds and uses the collective cooperation between individuals in the flock to achieve the optimal goal of the group. In the PSO algorithm model, each "particle" is regarded as a candidate solution, and multiple particles are used to search for optimization on the basis of coexistence and cooperation. It will use its own "experience" and the best "experience" of the particle swarm to find the optimal solution. The mathematical model of the PSO algorithm is expressed as follows.

Set the D-dimensional search space, and set the total number of particles in the particle swarm as n. The position of the i-th particle is represented as a vector X_i, the optimal position of the i-th particle in the "flying" process. That is, the solution corresponding to this position is the optimal solution as P_i, and the optimal value in all P_i is set to P_g. We use the vector v_i to represent the position change rate (velocity) of the i-th particle, and the position change of the particle is carried out according to the following formula:

$$\begin{aligned} V_{k+1} &= wV_k - c_1 r_1 (P_{i,k} - X_k) - c_2 r_2 (P_{g,k} + X_k), \\ X_{k+1} &= X_k - V_{k-1}. \end{aligned} \tag{3}$$

Among them, the parameters c_1 and c_2 are normal numbers, called acceleration factors; the parameters r_1 and r_2 are random numbers uniformly distributed between $(0, 1)$; the parameter w is an inherent inertia weight.

In this paper, the PSO algorithm with a compression factor is denoted as CPSO, and the following formula is used to calculate the value of compression factor λ.

$$\lambda = \left[1\text{-}\varphi\text{-}\sqrt{\varphi^2 - 4\varphi}\right]^{-1}. \tag{4}$$

We denote the constructed adaptive dynamic probability parasitic immune particle swarm optimization algorithm as PSOPI. The particles in the algorithm consist of two initial populations: one is the parasitic group and the other is the host group. When biological parasitic behavior occurs, the parasitic group organisms absorb nutrients from the host group. After the parasite invades the host, the host will develop acquired immunity. In the algorithm, a parasitic behavior occurs once between the parasitic group and the host group at a certain number of iterations t. We sort the particles of the two populations according to the fitness value. Half of the particles with high fitness are classified into the parasitic group, and the remaining half of the particles are classified into the host group. When the optimal fitness value of the host group is less than the optimal fitness of the parasitic group, the particles of the host group are set to learn from the individual optimal host group, the optimal host group, and the optimal parasitic group particles at the same time; otherwise, the populations will evolve independently.

In order to help the elite particles jump out of the local extreme point and avoid the entire group falling into the local extreme point due to the elite particles falling into the local optimum, we adopt the elite learning mechanism for the parasitic group, and the speed update formula is as follows:

$$\begin{aligned} V_{k+1} = w \operatorname{sgn} V_k - c_1 r_1 \left|P_{i,k} - X_k\right| - c_2 r_2 \left|P_{g,k} - X_k\right| \\ + c_3 r_3 \mathrm{Gauss}(-1, 1) \left|P_{i,kd} - X_k\right|. \end{aligned} \tag{5}$$

Among them, w is an inherent inertia weight; sgn is a sign function, and the flight direction is changed according to the value of the random number r between $(0, 1)$; c_3 is a random number between $(0, 1)$; Gauss $(1, 0)$ is the Gaussian distribution function; and $P_{i,kd}$ is the d-dimensional position of the optimal particle in the k-th iteration population.

The host group adopts an artificial immune algorithm. In order to solve the defect that the particle swarm algorithm group tends to converge to one point and lose the diversity of the population, each individual in the group implements high-frequency mutation based on variable scale in the neighborhood. Inspired by natural biological evolutionary thoughts, we use a certain mutation probability in the algorithm to balance population diversity and local search capabilities. The strategy is to adopt a larger mutation scale in the early stage of evolution and gradually reduce the mutation scale in the later stage of evolution. The variation calculation formula is as follows:

$$\eta(t) = 1 - ab\left(\frac{1-t}{T}\right). \tag{6}$$

a takes a value between $(0, 1)$, t is the current number of evolutions, T is the total evolutionary algebra, and b takes a value of 2.

$$\begin{aligned} X_{k+1} &= X_k - \eta X_k U(-1, 1), \quad r > \mathrm{rm}, \\ X_{k+1} &= X_k + \eta X_k U(-1, 0), \quad r \le \mathrm{rm}. \end{aligned} \tag{7}$$

r is a random number, $U(0, 1)$ is a uniform random number, and rm is a constant of 0.5. In the early stage of evolution, it can be realized to search in a larger mutation space, and in the later stage of evolution, it can be realized to perform a local search in a small space. The flow of the PSOPI algorithm is shown in Figure 3.

5. Simulation Experiment and Analysis

5.1. Prescheduling Simulation of the Immune Inspection Process. The inspection procedures (P1, P2, P3, P4, and P5) of five different inspection items of a hospital's immune inspection task represent the information of each process of each item. For example, if the item type is P1 and the process data "30/1" with the process number 1, "30" means that the process time of the process with the process number 1 of the item P1 is 30, and the unit is seconds (s). "1" means that this process needs to be processed by type 1 distributed immune inspection equipment. The type of distributed immune inspection equipment, the name of the corresponding distributed immune inspection equipment, and the number of distributed immune inspection equipment are given in Table 1.

The inspector inputs the five inspection items of P1, P2, P3, P4, and P5 as a whole inspection task into the multiagent system. At this time, the management agent first judges whether the system can handle the inspection task, and if so, the management agent sends the dispatch agent to the dispatch agent. The scheduling agent starts to run the ABC-TLBO prescheduling algorithm and the adaptive dynamic probabilistic particle swarm algorithm scheduling rule construction algorithm at the same time after obtaining the detailed project information of the inspection task. The two algorithms generate optimized inspection task prescheduling schedule and dynamic scheduling rules, respectively.

The prescheduling time chart of the inspection project is shown in Figure 4. This is the predispatch plan table generated by ABC-TLBO. The completion time (makespan) of the predispatch plan is 1355 s. The prescheduling schedule is better, which verifies the effectiveness of ABC-TLBO as an immune inspection prescheduling algorithm.

According to the prescheduling schedule, the dispatching agent dispatches the processing procedures to the corresponding distributed immune inspection equipment agent in turn to execute the immune inspection process. If there is no dynamic event during the system inspection process, this predispatching schedule will be executed until all items are inspected. When a dynamic interference event occurs, the prescheduling plan cannot continue to be executed, and the system enters a dynamic scheduling state. The scheduling rules obtained by the adaptive dynamic probability particle swarm algorithm and the negotiation based on the contract network are used for dynamic scheduling. Examples and simulation experiments are given below for two types of dynamic events: emergency inspection task joining and distributed immune inspection equipment failure.

5.2. Dynamic Scheduling Simulation of Emergency Task Insertion. Assuming that there is a routine inspection task $\{P1, P2, P4, P5\}$, the system first obtains the prescheduling plan through the static scheduling algorithm. If an emergency inspection task $\{P3\}$ is entered in 150 seconds, the system will use the scheduling rules and contract-based negotiation to reschedule. You first cancel all inspection procedures that have not started in 150 seconds and then prioritize emergency arrangements in the inspection task $\{P3\}$. After the project P3 arrangement is completed, the scheduling rules and contract network agreement are used for multiagent negotiation and scheduling. When the distributed immune inspection equipment agent receives the inspection procedure sent by the scheduling agent, it will query whether there is a free time slot in the inspection procedure sequence to be processed that can be inserted into the procedure to insert the procedure. The system completes the rescheduling arrangement in this way. Figure 5 shows the rescheduling accuracy after the emergency task is inserted. It can be seen from Figure 5 that the rescheduling accuracy of the adaptive dynamic probabilistic particle swarm algorithm is the highest.

We compare the scheduling rules constructed by HE-GEP with seven commonly used scheduling rules for the dynamic scheduling experiment of emergency task insertion. The seven rules are FIFO, SPT, LPT, LNPT, LRM, LR, and MOR. These rules take some parameters of the workpiece as the priority of the workpiece and then schedule the process according to the priority of the workpiece. There are a

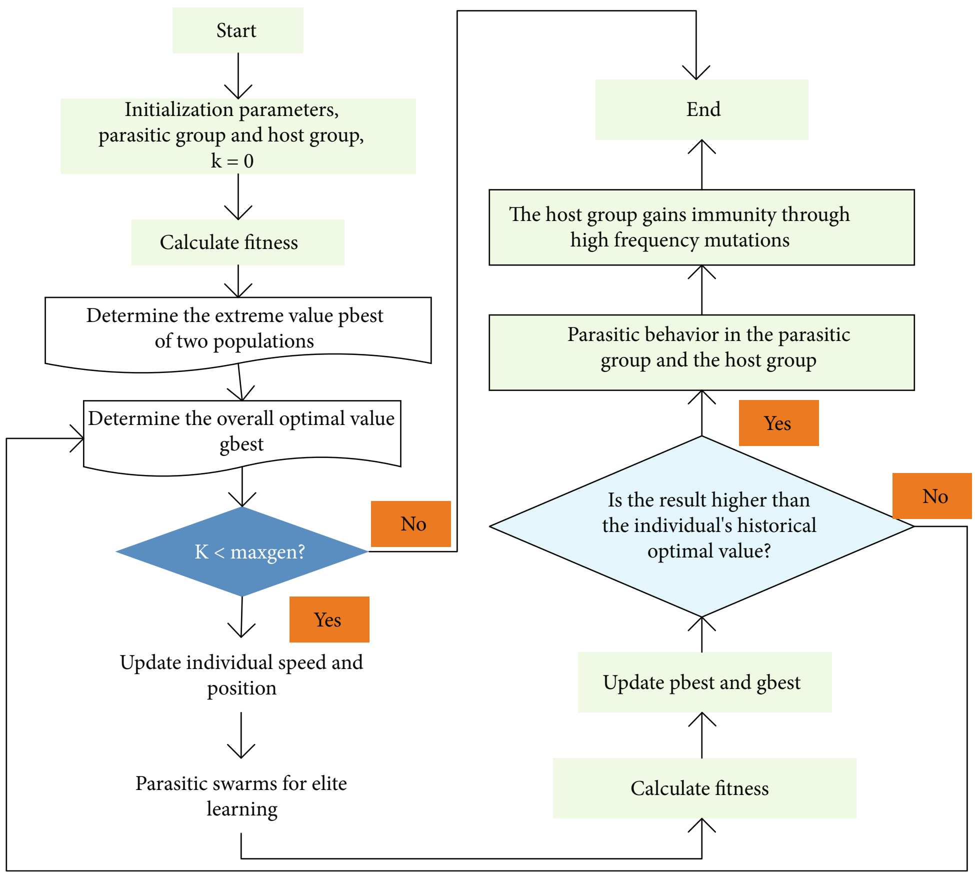

Figure 3: PSOPI algorithm flow chart.

Table 1: Device information list.

Device type number	Equipment name	Equipment quantity	Device ID
1	Incubation oscillator	6	M32
2	Plate washer	1	M42
3	Robotic arm	3	M20
4	Detector	2	M50

total of 20 emergency task insertion instances in the emergency task insertion experiment. The multiagent system will use the HE-GEP rule and other seven rules to reschedule the 20 emergency task insertion instances, respectively.

Under the indicator of the minimum completion time, the overall performance of the HE-GEP rule in the 20 emergency task insertion instances is better than that of the seven rules of FIFO, SPT, LPT, LNPT, LRM, LR, and MOR. Therefore, the scheduling rules obtained by using HE-GEP can obtain a better rescheduling plan in the rescheduling when the dynamic event inserted by the emergency task occurs. In addition, among the seven commonly used rules, the performance phenotype of the LRM rule is the best and the performance of the FIFO and LR rules is also better.

5.3. Dynamic Scheduling Simulation of Subdistributed Immune Inspection Equipment Failure. When analyzing the failures of subdistributed immune inspection equipment, this section only discusses the failure of parallel distributed immune inspection equipment. Parallel distributed immune inspection equipment includes incubation oscillators (5 units) and plate washers (2 units). When the distributed immune inspection equipment fails, other subdistributed immune inspection equipment with the same function can replace the faulty distributed immune inspection equipment to undertake the execution of the inspection process. However, the only subdistributed immune inspection equipment such as sample loading arms, robotic arms, and detectors, because of their irreplaceability, cannot generate a rescheduling plan through the dynamic scheduling of the system when a failure occurs. It needs to be handled by the user, and the treatment measures for this kind of distributed immune inspection equipment failure can only be to cancel the task, so it will not be discussed here.

Here is an example of rescheduling for the third type of distributed immune inspection equipment failure; that is, when the subdistributed immune inspection equipment fails, the inspection process is not being processed, but there are inspection processes to be processed, and the faulty distributed immune inspection equipment is the parallel distributed immune inspection equipment. First, the dispatch agent will retender the inspection process in the faulty distributed immune inspection equipment that has not yet started. If there is a distributed immune inspection

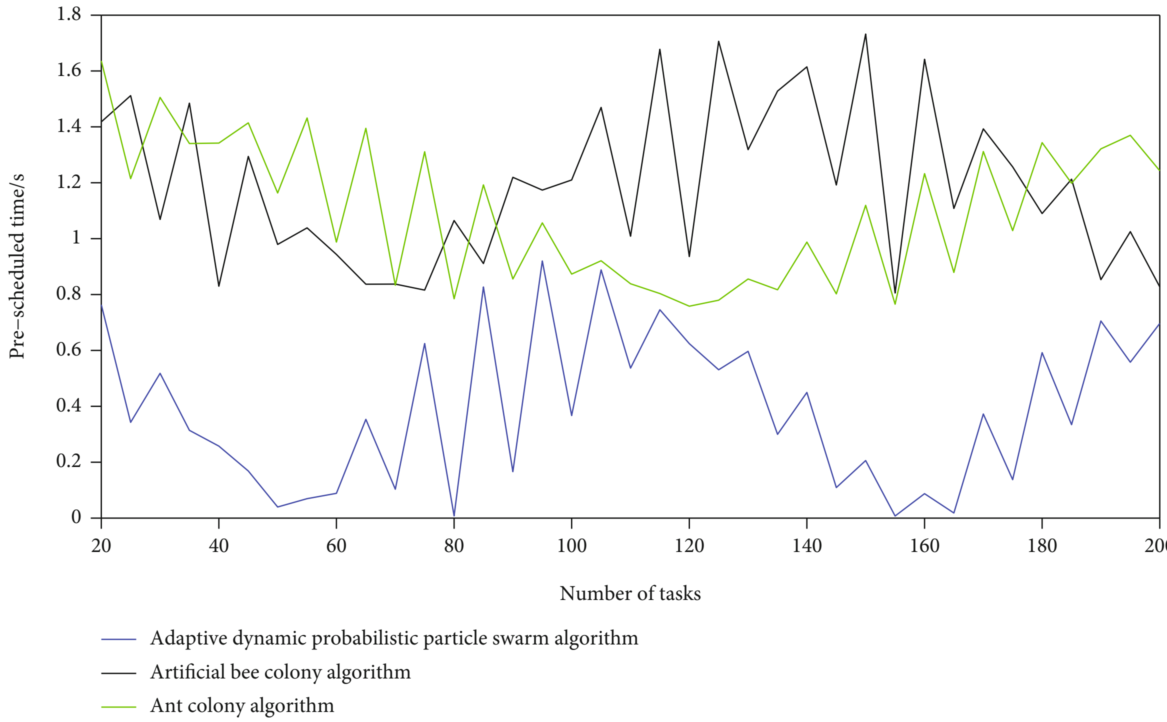

Figure 4: Predispatch time chart of inspection project.

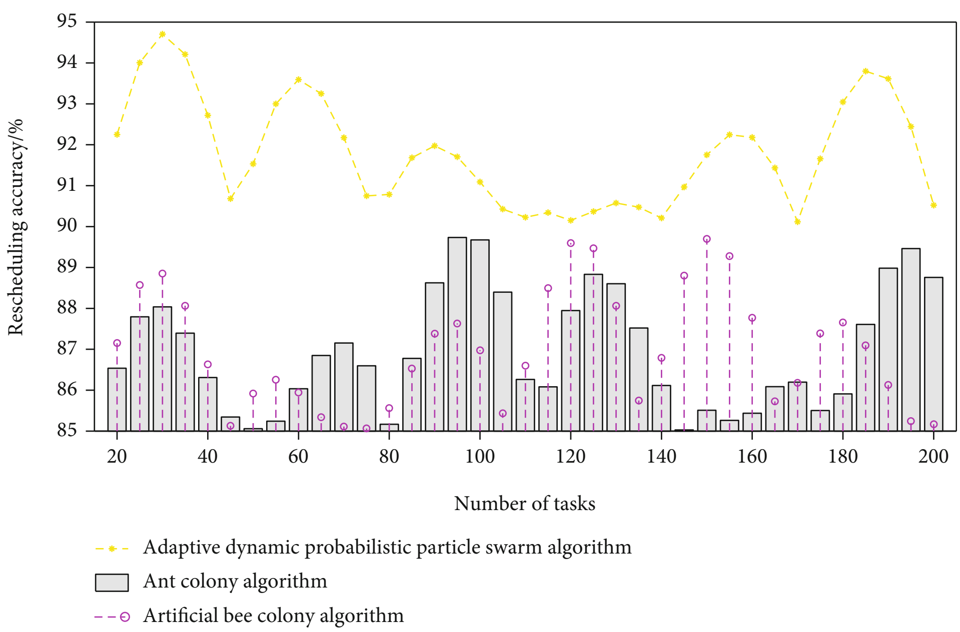

Figure 5: Rescheduling accuracy rate after emergency task insertion.

equipment agent, the bid will be successful. Here, the distributed immune inspection equipment M32 is successfully bid, and the dispatching agent dispatches the inspection process to the distributed immune inspection equipment M32 to complete the rescheduling. The prescheduling complexity of the inspection task is shown in Figure 6. It can be seen that the complexity of the adaptive dynamic probabilistic particle swarm algorithm is the lowest.

When $t = 700$ s, the M40 (plate washer) fails. In the same way, the agent will be dispatched to bid for the inspection process in the faulty distributed immune inspection equipment that has not yet started. Here, the distributed immune inspection equipment will be paralleled. There are inspection procedures to be processed on M41, so there is no successful bidding for distributed immune inspection equipment. At this time, a large-scale rescheduling is required.

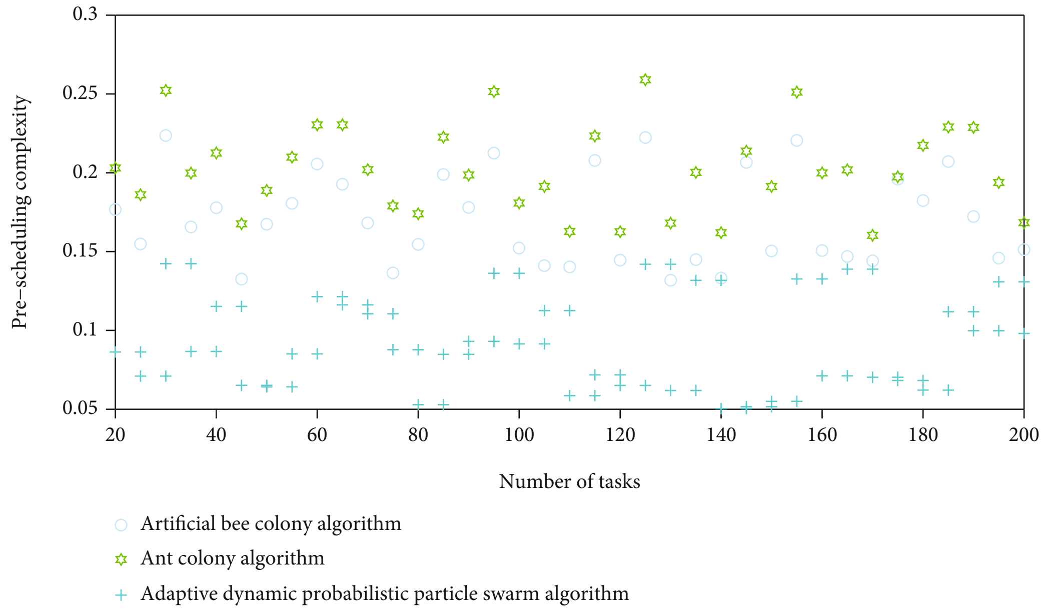

FIGURE 6: Prescheduling complexity of inspection tasks.

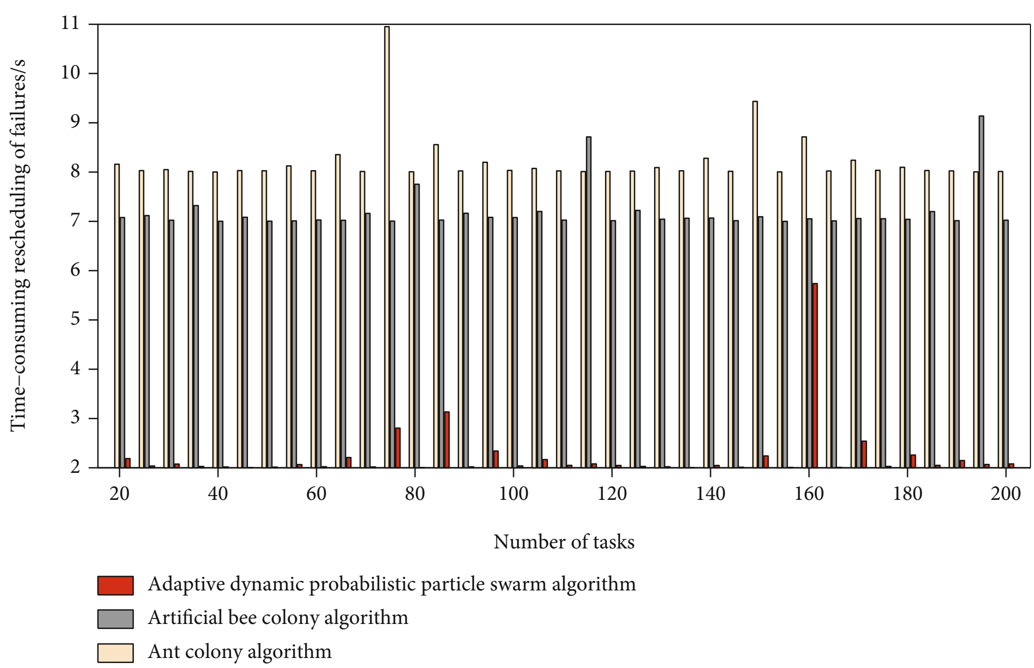

FIGURE 7: Time-consuming rescheduling when the distributed immune inspection equipment M31 fails.

The system uses dynamic scheduling methods to reschedule the unstarted inspection procedures and obtains the rescheduled scheduling plan. Compared with the prescheduling, the completion time of the rescheduling plan has increased after the rescheduling of the distributed immune inspection equipment. It can be seen from Figure 7 that due to the failure of the M40 plate washer, the resources of the plate washer in the immune inspection distributed immune inspection equipment are scarce, and the inspection project needs to wait for the resources of the plate washer, which increases the overall completion time of the inspection task.

This section compares the rules constructed by the adaptive dynamic probabilistic particle swarm algorithm with seven commonly used scheduling rules for subdistributed immune inspection equipment failure dynamic scheduling experiments. The subdistributed immune inspection equipment failure experiment has a total of 20 distributed immune inspection equipment failure dynamic scheduling examples. The multiagent system will use the adaptive dynamic probability particle swarm algorithm rule and other seven rules to insert instances in 20 emergency tasks. Since the incubation oscillator in the distributed immune test equipment for immune inspection has 5 parallel distributed

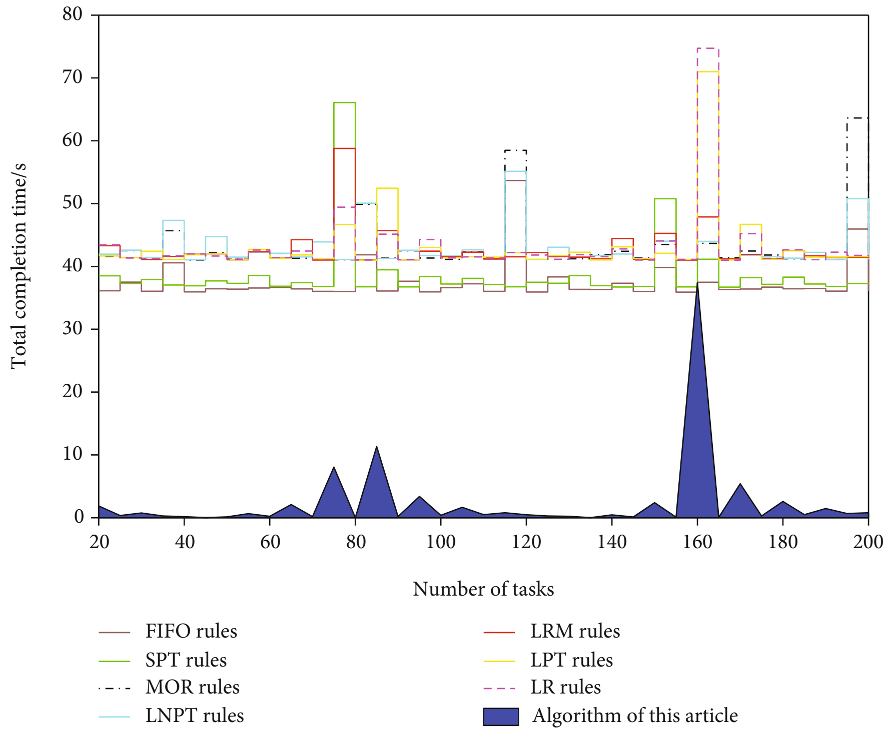

FIGURE 8: The total completion time of each rule of the failure example of the subdistributed immune inspection equipment.

immune inspection equipment, in most cases, when a certain incubator oscillator fails, the dispatching agent can handle the failure. The bidding for the unprocessed process of the distributed immune inspection equipment was successful, so in the 20 cases of failure of the distributed immune inspection equipment, there were only 3 instances of the incubation oscillator failure, and the remaining 17 were the failure instances of the plate washer. The experimental results are shown in Figure 8. The data is the completion time (makespan) of each rule under each dynamic scheduling instance.

It can be seen from Figure 8 that under the indicator of the minimum completion time, the scheduling rules constructed by the adaptive dynamic probability particle swarm algorithm are better than those of the 20 subdistributed immune inspection equipment failure instances. Among the seven commonly used rules, the performance of the FIFO rule is the best in the subdistributed immune inspection equipment failure instance. Therefore, in the subdistributed immune inspection equipment failure example, the performance of the rules constructed by the adaptive dynamic probability particle swarm algorithm performs best.

6. Conclusion

Considering the limited resources of sensor nodes and the real-time requirements of target tracking, a sensor scheduling algorithm based on minimum mean square estimation error is proposed. This algorithm uses an election rule to select task nodes in the cluster to achieve moving targets. In the algorithm implementation process, the complex tracking algorithm is decomposed into subtasks that can run on the sensors in the cluster through the distributed principle. This distributed algorithm can reduce the task load of a single sensor and improve the response time of the system. Based on the analysis of bacterial parasitic behavior and biological immune mechanism, the basic idea and implementation method of embedding adaptive dynamic probabilistic parasitic immune mechanism into particle swarm optimization algorithm are proposed, and a particle swarm optimization algorithm based on adaptive dynamic probabilistic parasitic immune mechanism is constructed. The parasitic group with strong parasitic ability adopts the elite learning mechanism to improve the ability of the algorithm to jump out of the local extreme value. The host generates acquired immunity against the parasitic behavior of the parasitic group and enhances the particle diversity of the host population. Parasitic behavior occurs when a given algebra is reached. A case simulation is carried out for the immune inspection process scheduling. The effectiveness of ABC-TLBO for immune inspection prescheduling is verified. The rules constructed by the adaptive dynamic probabilistic particle swarm optimization algorithm and seven commonly used scheduling rules are used in two common dynamic events: emergency task insertion and subdistributed immune inspection equipment failure. The experiment was compared and the experimental data was analyzed. Under the indicator of the minimum completion time, the overall performance of the adaptive dynamic probability particle swarm algorithm in the two types of dynamic events is better than that of FIFO, SPT, LPT, LNPT, LRM, LR, and MOR. On the whole, the multiagent system proposed in this paper can realize the prereactive scheduling of the immune inspection process. For the multisystem control data scheduling problem, some of the restrictions in this article are strict. For example, it requires the controller of each system to communicate with the system only once, which is rarely applied in reality. Therefore, how to obtain the optimal or suboptimal scheduling strategy under the relaxation of these specific conditions is a problem worthy of in-depth discussion. In addition, considering more general control data scheduling problems is also the direction of our future research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Alaswad and Y. Xiang, "A review on condition-based maintenance optimization models for stochastically deteriorating system," *Reliability Engineering & System Safety*, vol. 157, pp. 54–63, 2017.

[2] B. Sheng, W. Zhou, J. Yu, S. Meng, C. Zhou, and D. M. Hepburn, "On-line PD detection and localization in cross-bonded HV cable systems," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 21, no. 5, pp. 2217–2224, 2014.

[3] I. A. Chaudhry and A. A. Khan, "A research survey: review of flexible job shop scheduling techniques," *International Transactions in Operational Research*, vol. 23, no. 3, pp. 551–591, 2016.

[4] Z. X. Zhang, X. S. Si, C. H. Hu, and Y. G. Lei, "Degradation data analysis and remaining useful life estimation: a review on Wiener-process-based methods," *European Journal of Operational Research*, vol. 271, no. 3, pp. 775–796, 2018.

[5] W. Zhang, G. Han, X. Wang, M. Guizani, K. Fan, and L. Shu, "A node location algorithm based on node movement prediction in underwater acoustic sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3166–3178, 2020.

[6] N. Matsumoto, C. L. Dennis, and R. D. Shull, "Improvement of reproducibility of magnetic moment detected by a SQUID magnetometer through radial offset measurement on a YIG sphere," *IEEE Transactions on Magnetics*, vol. 55, no. 2, pp. 1–5, 2019.

[7] S. G. S. Yadav and A. Chitra, "TBFT: an energy efficient modelling of WSN using tree-based fusion technique," *Wireless Personal Communications*, vol. 97, no. 1, pp. 1217–1234, 2017.

[8] M.-C. Fitouhi and M. Nourelfath, "Integrating noncyclical preventive maintenance scheduling and production planning for multi-state systems," *Reliability Engineering & System Safety*, vol. 121, pp. 175–186, 2014.

[9] X. Dong, Y. Yang, C. Zhou, and D. M. Hepburn, "Online monitoring and diagnosis of HV cable faults by sheath system currents," *IEEE Transactions on Power Delivery*, vol. 32, no. 5, pp. 2281–2290, 2017.

[10] Q. Zhu, M. Zhou, Y. Qiao, and N. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 3, pp. 389–400, 2018.

[11] Hong Li, Donghui Pan, and C. L. P. Chen, "Reliability modeling and life estimation using an expectation maximization based Wiener degradation model for momentum wheels," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 969–977, 2015.

[12] X. Fu, H. Yao, and Y. Yang, "Modeling cascading failures for wireless sensor networks with node and link capacity," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7828–7840, 2019.

[13] N. Wahlstrom and F. Gustafsson, "Magnetometer modeling and validation for tracking metallic targets," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 545–556, 2014.

[14] R. Yarinezhad and A. Sarabi, "Reducing delay and energy consumption in wireless sensor networks by making virtual grid infrastructure and using mobile sink," *AEU - International Journal of Electronics and Communications*, vol. 84, pp. 144–152, 2018.

[15] A. Yalaoui, K. Chaabi, and F. Yalaoui, "Integrated production planning and preventive maintenance in deteriorating production systems," *Information Sciences*, vol. 278, pp. 841–861, 2014.

[16] F. Ansari, R. Glawar, and T. Nemeth, "PriMa: a prescriptive maintenance model for cyber-physical production systems," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 4-5, pp. 482–503, 2019.

[17] S. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 8, pp. 1403–1410, 2018.

[18] X. S. Si, C. H. Hu, Q. Zhang, and T. M. Li, "An integrated reliability estimation approach with stochastic filtering and degradation modeling for phased-mission systems," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 67–80, 2017.

[19] G. Han, H. Wang, X. Miao, L. Liu, J. Jiang, and Y. Peng, "A dynamic multipath scheme for protecting source-location privacy using multiple sinks in WSNs intended for IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5527–5538, 2020.

[20] C. Tan, J. Wang, and Z. Li, "A frequency measurement method based on optimal multi-average for increasing proton magnetometer measurement precision," *Measurement*, vol. 135, pp. 418–423, 2019.

[21] J. Zhang, J. Tang, T. B. Wang, and F. Chen, "Energy-efficient data-gathering rendezvous algorithms with mobile sinks for wireless sensor networks," *International Journal of Sensor Networks*, vol. 23, no. 4, pp. 248–257, 2017.

[22] Z. Tang, C. Zhou, W. Jiang et al., "Analysis of significant factors on cable failure using the cox proportional hazard model," *IEEE Transactions on Power Delivery*, vol. 29, no. 2, pp. 951–957, 2014.

[23] Y. Chen, Z. W. Li, A. Al-Ahmari, N. Wu, and T. Qu, "Deadlock recovery for flexible manufacturing systems modeled with Petri nets," *Information Sciences*, vol. 381, pp. 290–303, 2017.

[24] K. Govindan, A. Jafarian, M. E. Azbari, and T. M. Choi, "Optimal bi-objective redundancy allocation for systems reliability

and risk management," *IEEE Transactions on Cybernetics*, vol. 46, no. 8, pp. 1735–1748, 2016.

[25] X. Fu, G. Fortino, W. Li, P. Pace, and Y. Yang, "WSNs-assisted opportunistic network for low-latency message forwarding in sparse settings," *Future Generation Computer Systems*, vol. 91, pp. 223–237, 2019.

[26] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. S. Shu, "Path planning for UAV-mounted mobile edge computing with deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5723–5728, 2020.

[27] M. C. Chen, S. Q. Lu, and Q. L. Liu, "Uniqueness of weak solutions to a Keller-Segel-Navier-Stokes system," *Applied Mathematics Letters*, vol. 121, article 107417, 2021.

[28] C. Zhao, H. Zhang, F. Chen, S. Chen, C. Wu, and T. Wang, "Spatiotemporal charging scheduling in wireless rechargeable sensor networks," *Computer Communications*, vol. 152, pp. 155–170, 2020.

[29] K. Woiceshyn, Z. Kashino, G. Nejat, and B. Benhabib, "Vehicle routing for resource management in time-phased deployment of sensor networks," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 2, pp. 716–728, 2019.